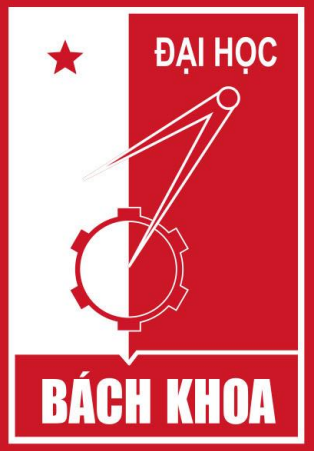


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhập môn Công nghệ Phần mềm

(Introduction to Software Engineering)

CHƯƠNG 6

Kỹ nghệ yêu cầu phần mềm (Requirement Engineering)

Mục tiêu của bài học

Sinh viên sẽ được trang bị các kiến thức sau:

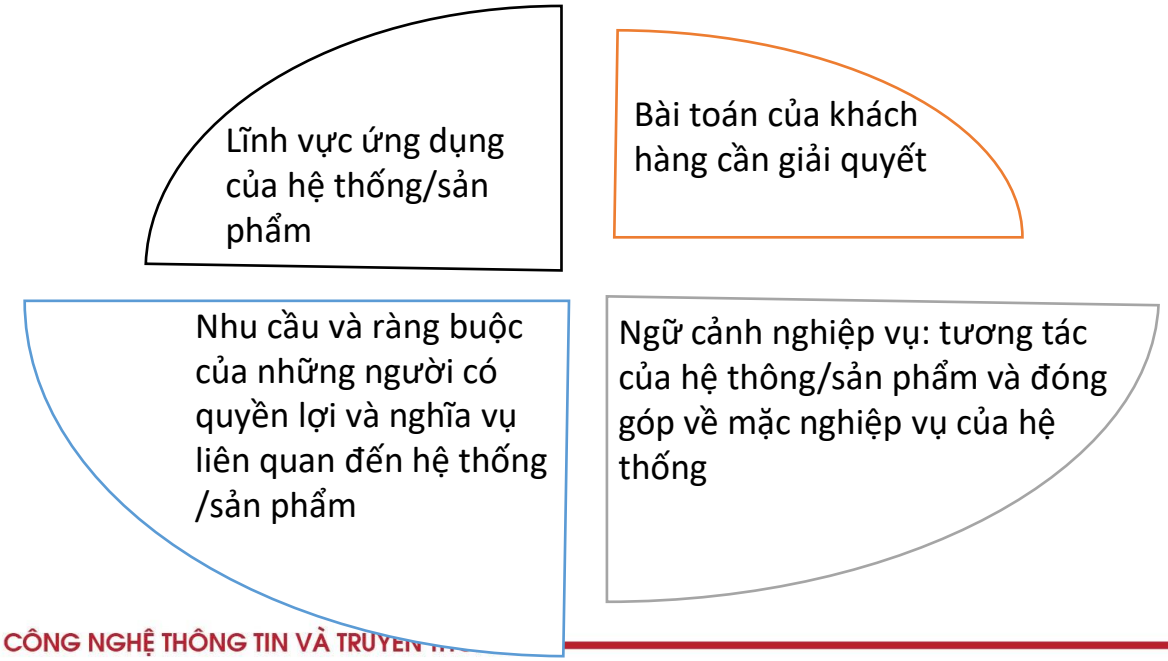
- Hiểu rõ các khái niệm liên quan tới kĩ nghệ yêu cầu phần mềm
- Biết, nắm vững vị trí, vai trò của kĩ nghệ YCPM
- Có khái niệm về một số yếu tố liên quan tới các yêu cầu chức năng và phi chức năng
- Nắm vững các hoạt động chính của kĩ nghệ YCPM

Nội dung

1. Khái niệm
2. Tầm quan trọng của yêu cầu phần mềm
3. Yêu cầu chức năng và yêu cầu phi chức năng
4. Các hoạt động chính trong kỹ nghệ yêu cầu phần mềm

1. Khái niệm (1)

- Các đặc tính của hệ thống hay sản phẩm do khách hàng - người sử dụng PM - đặt ra → Xác định được phần mềm đáp ứng được các yêu cầu và mong muốn của khách hàng - người sử dụng phần mềm



1. Khái niệm (2)

- Khởi đầu (Inception): Hỏi một loạt các câu hỏi để xác định:
  - Hiểu biết căn bản về vấn đề cần giải quyết.
  - Người đang cần giải pháp
  - Loại giải pháp mong muốn
  - Mức độ hiệu quả ban đầu của việc trao đổi thông tin giữa khách hàng và nhà phát triển
- Khám phá (Elicitation): tìm ra yêu cầu của tất cả khách hàng.
- Xây dựng (Elaboration): tạo ra mô hình phân tích xác định dữ liệu, chức năng và hành vi được yêu cầu.
- Đàm phán (Negotiation): đồng ý với một hệ thống có thể bàn giao một cách thực tế đối với cả 2 bên.

1. Khái niệm (3)

- Đặc tả (Specification): có thể là một/ nhiều những thứ sau:
  - Một Tài liệu được viết
  - Một Tập hợp các mô hình
  - Một hình thức biểu diễn toán học
  - Một tập các kịch bản người dùng (use-case)
  - Một Nguyên mẫu
- Đánh giá (Validation): tạo cơ chế xem xét các vấn đề:
  - Sai sót trong nội dung hoặc giải thích.
  - Phần được yêu cầu làm rõ.
  - Thông tin bị thiếu
  - Mâu thuẫn
  - Yêu cầu không thực tế, không thể đạt được.
- Quản lý các Yêu cầu (Requirements management)

Nội dung

1. Khái niệm
2. **Tầm quan trọng của yêu cầu phần mềm**
3. Yêu cầu chức năng và yêu cầu phi chức năng
4. Các hoạt động chính trong kỹ nghệ yêu cầu phần mềm

Nguồn gốc yêu cầu phần mềm

- **Người sử dụng** (Khách hàng): theo mô hình phân lớp của yêu cầu phần mềm Khách hàng được chia làm hai loại:
  - Khách hàng cung cấp các business requirement: cung cấp các thông tin về công ty, về các đặc điểm ở mức độ cao, về mô hình và phạm vi của hệ thống
  - Khách hàng cung cấp các user requirement: cung cấp các công tin về từng nhiệm vụ cụ thể mà họ sẽ làm việc với phần mềm
- Cần phải phối hợp, kết hợp chặt chẽ với hai phân loại khách hàng trên

Đặc điểm của khách hàng phần mềm

- Khách hàng chỉ có những ý tưởng còn mơ hồ về phần mềm cần phải xây dựng để phục vụ công việc của họ.
- Cho nên chúng ta phải sẵn sàng, kiên trì theo đuổi để đi từ các ý tưởng mơ hồ đó đến "Phần mềm có đầy đủ các tính năng cần thiết"
- Khách hàng rất hay thay đổi các đòi hỏi của mình, chúng ta nắm bắt được các thay đổi đó và sửa đổi các mô tả một cách hợp lý

Vấn đề YCPM giải quyết (1)

- Sự tham gia quá mức của NSD
  - ✓ Thông thường người sử dụng không hiểu rõ về quá trình xây dựng các yêu cầu phần mềm và các đặc điểm của phần mềm.
  - ✓ Họ sẽ đưa những đòi hỏi quá cao hoặc chẳng liên quan đến quá trình phát triển phần mềm như viết code, ...
  - ✓ Họ đưa ra những yêu cầu và đề nghị rất khó chấp nhận và gây khó khăn cho các PTV
- Có quá nhiều yêu cầu trong yêu cầu phần mềm
  - ✓ Thông thường các yêu cầu phần mềm được phát hiện trong quá trình khảo sát và rất có thể các yêu cầu về phần mềm sẽ lớn hơn khả năng của đội ngũ phát triển về: nhân lực, thời gian, tài chính.
  - ✓ Cần hạn chế không để các yêu cầu phần mềm phát sinh đi quá phạm vi và giới hạn của phần mềm
  - ✓ Cần quản lý các thay đổi về yêu cầu phần mềm một cách hợp lý và xem xét ảnh hưởng của nó tới kiến trúc hệ thống, … trong quá trình phát triển

Vấn đề YCPM giải quyết (2)

- Các yêu cầu phần mềm mơ hồ nhập nhằng
  - ✓ Đây là một vấn đề rất hay xảy ra trong quá trình phát triển các yêu cầu phần mềm
  - ✓ Các yêu cầu phần mềm cần phải rõ ràng, không được phép hiểu theo nhiều cách
  - ✓ Phương pháp sửa các yêu cầu mơ hồ nhập nhằng là làm lại, đặc tả lại các yêu cầu này.
  - ✓ Theo đánh giá của các nhà phân tích: làm lại yêu cầu phần mềm thường chiếm khoảng 40% quá trình xây dựng nó và 70-80% các đặc tính xây dựng lại có thể dẫn đến các lỗi
  - ✓ Lưu ý tới từng yêu cầu phần mềm và không để sót những yêu cầu mơ hồ, không rõ ràng
- Không lưu ý tới người sử dụng phần mềm
  - ✓ Thông thường phần mềm làm ra cho một tập hợp các đôi tượng nào đó sử dụng
  - ✓ Cần quan tâm tới đặc điểm của đối tượng này

Vấn đề YCPM giải quyết (3)

- Các đặc tính thừa:
  - ✓ Thông thường người phát triển theo các thói quen nghề nghiệp thêm vào các yêu cầu phần mềm các chức năng không cần thiết cho phần mềm
  - ✓ Tương tự như vậy người sử dụng có thể đưa ra một số yêu cầu phụ cho phần mềm. Các yêu cầu này có thể đòi hỏi các tốn kém về mặt xây dựng mà trên thực tế hoàn toàn không cần thiết
  - ✓ Cần đánh giá các đặc tính và tính cần thiết của nó đối với các phần mềm. Những đặc tính phụ có thể xem xét kỹ hơn xem khả năng đáp ứng nó về mặt kỹ thuật có đáng giá hay không.
- Đặc tả quá ít
  - ✓ NSD là chuyên viên trong một lĩnh vực nào đó có thói quen nghĩ rằng tất cả các LTV đều là các chuyên viên trong lĩnh vực đó.
  - ✓ NSD đưa ra những yêu cầu quá ngắn gọn mà không miêu tả kỹ lưỡng chúng là gì
  - ✓ Cần hỏi rõ NSD và tranh thủ các kiến thức của họ

Vấn đề YCPM giải quyết (4)

- Kế hoạch sai:
  - ✓Các LTV đánh giá sai về mức độ phức tạp của vấn đề và dẫn tới lập kế hoạch sai về mặt thời gian hoàn thành.
  - ✓Cần lưu ý đánh giá thật chắc chắn các đặc tính của phần mềm khi lập kế hoạch xây dựng phần mềm
  - ✓Nên trả lời cho khách hàng các câu trả lời dạng "gần chính xác": trong trường hợp xấu nhất…., nếu…. Thì.

Nội dung

1. Khái niệm
2. Tầm quan trọng của yêu cầu phần mềm
3. **Yêu cầu chức năng và yêu cầu phi chức năng**
4. Các hoạt động chính trong kỹ nghệ yêu cầu phần mềm

Phân loại yêu cầu

- Theo 4 thành phần của phần mềm:
  - Các yêu cầu về phần mềm (Software)
  - Các yêu cầu về phần cứng (Hardware)
  - Các yêu cầu về dữ liệu (Data)
  - Các yêu cầu về con người (People, Users)
- Theo cách đặc tả phần mềm
  - Các yêu cầu chức năng
  - Các yêu cầu ngoài chức năng
  - Các ràng buộc khác

Yêu cầu chức năng

- Miêu tả các chức năng của hệ thống, phụ thuộc vào kiểu phần mềm và mong đợi của người dùng
  - Tương tác giữa phần mềm và môi trường, độc lập với việc cài đặt
  - Ví dụ: Hệ thống đồng hồ phải hiển thị thời gian dựa trên vị trí của nó
- Các công cụ đặc tả yêu cầu chức năng tiêu biểu:
  - Biểu đồ luồng dữ liệu (Data Flow Diagrams)
  - Máy trạng thái hữu hạn (Finite State Machines)
  - Mạng Petri (Petri nets),…
  - Tuy nhiên không bắt buộc và có thể dùng ngôn ngữ tự nhiên.

Yêu cầu phi chức năng và ràng buộc

- Yêu cầu phi chức năng: Định nghĩa các khía cạnh sử dụng phần mềm, không liên quan trực tiếp tới các hành vi chức năng:
  - Các tính chất của hệ thống như độ tin cậy, thời gian trả lời, dung lượng bộ nhớ, …
    - Thời gian trả lời phải nhỏ hơn 1 giây
- Ràng buộc: do khách hàng hay môi trường thực thi phần mềm đặt ra
  - Các yêu cầu do tổ chức qui định như qui định chuẩn về quá trình tiến hành, chuẩn tài liệu, …
    - Ngôn ngữ cài đặt phải là COBOL
  - Các yêu cầu từ bên ngoài
    - Phải giao tiếp với hệ thống điều phối được viết vào năm 1956.
- Thường sử dụng các công cụ
  - Biểu đồ thực thể liên kết (Entity-Relationship Diagrams)
  - Đặc tả Logic (Logic Specifications)
  - Đặc tả đại số (Algebraic Specifications)

→ Khó phát biểu chính xác, Rất khó kiểm tra

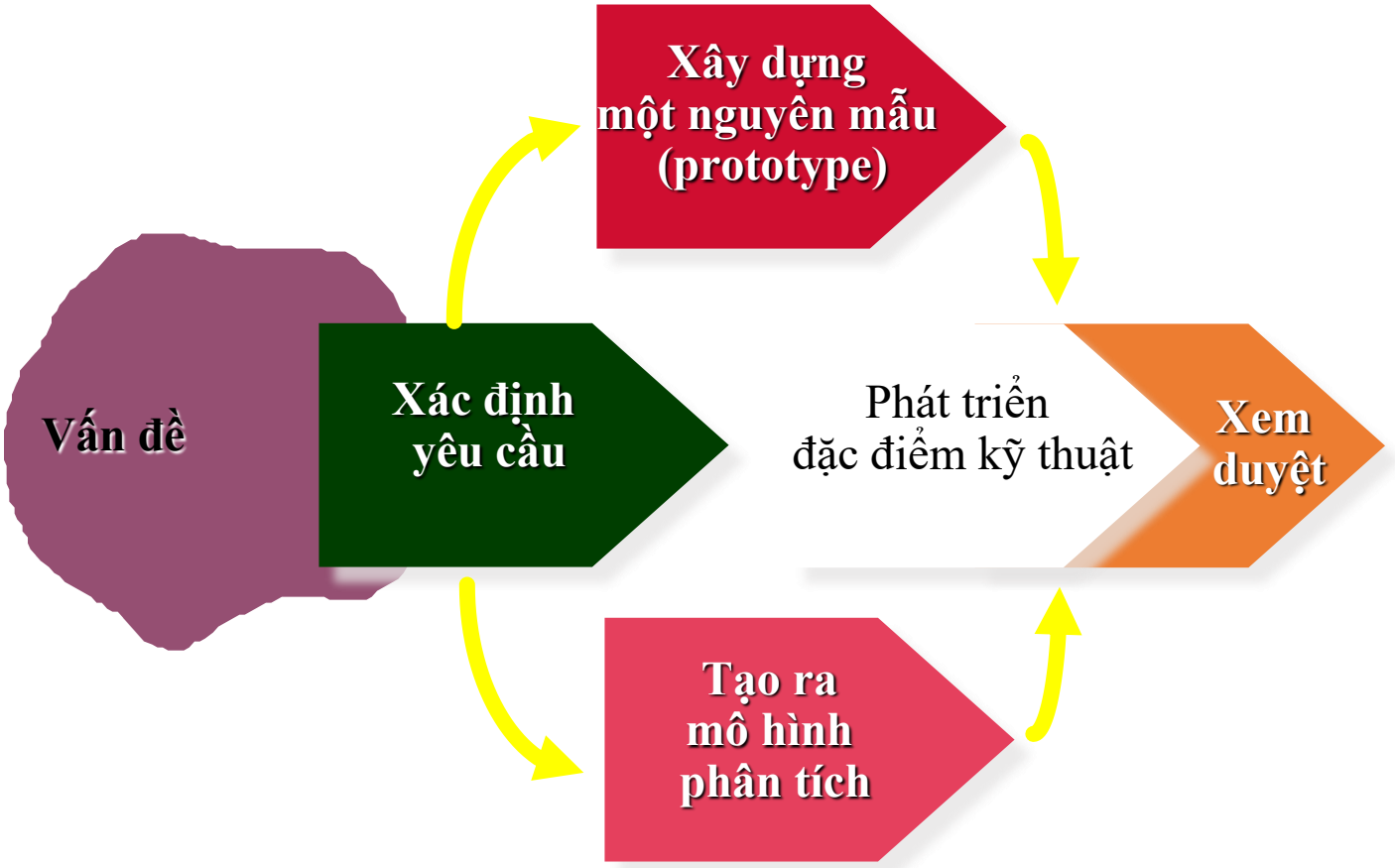
Nội dung

1. Khái niệm
2. Tầm quan trọng của yêu cầu phần mềm
3. Yêu cầu chức năng và yêu cầu phi chức năng
4. **Các hoạt động chính trong kỹ nghệ yêu cầu phần mềm**

4. Các hoạt động chính trong kỹ nghệ YCPM

- Phát hiện các yêu cầu phần mềm (Requirements elicitation)
- Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (Requirements analysis and negotiation)
- Đặc tả các yêu cầu phần mềm (Requirements specification)
- Mô hình hóa hệ thống (System modeling)
- Kiểm tra tính hợp lý của các yêu cầu phần mềm (Requirements validation)
- Quản trị các yêu cầu phần mềm (Requirements management)

4. Các hoạt động chính (tiếp)



4a. Phát hiện yêu cầu phần mềm

- Đánh giá tính khả thi về kỹ thuật và nghiệp vụ của phần mềm định phát triển
- Tìm kiếm các nhân sự (chuyên gia, người sử dụng) có những hiểu biết sâu sắc nhất, chi tiết nhất về hệ thống giúp chúng ta xác định yêu cầu phần mềm
- Xác định môi trường kỹ thuật trong đó sẽ triển khai phần mềm
- Xác định các ràng buộc về lĩnh vực ứng dụng của phần mềm (giới hạn về chức năng/hiệu năng phần mềm)

4a.Phát hiện yêu cầu phần mềm (tiếp)

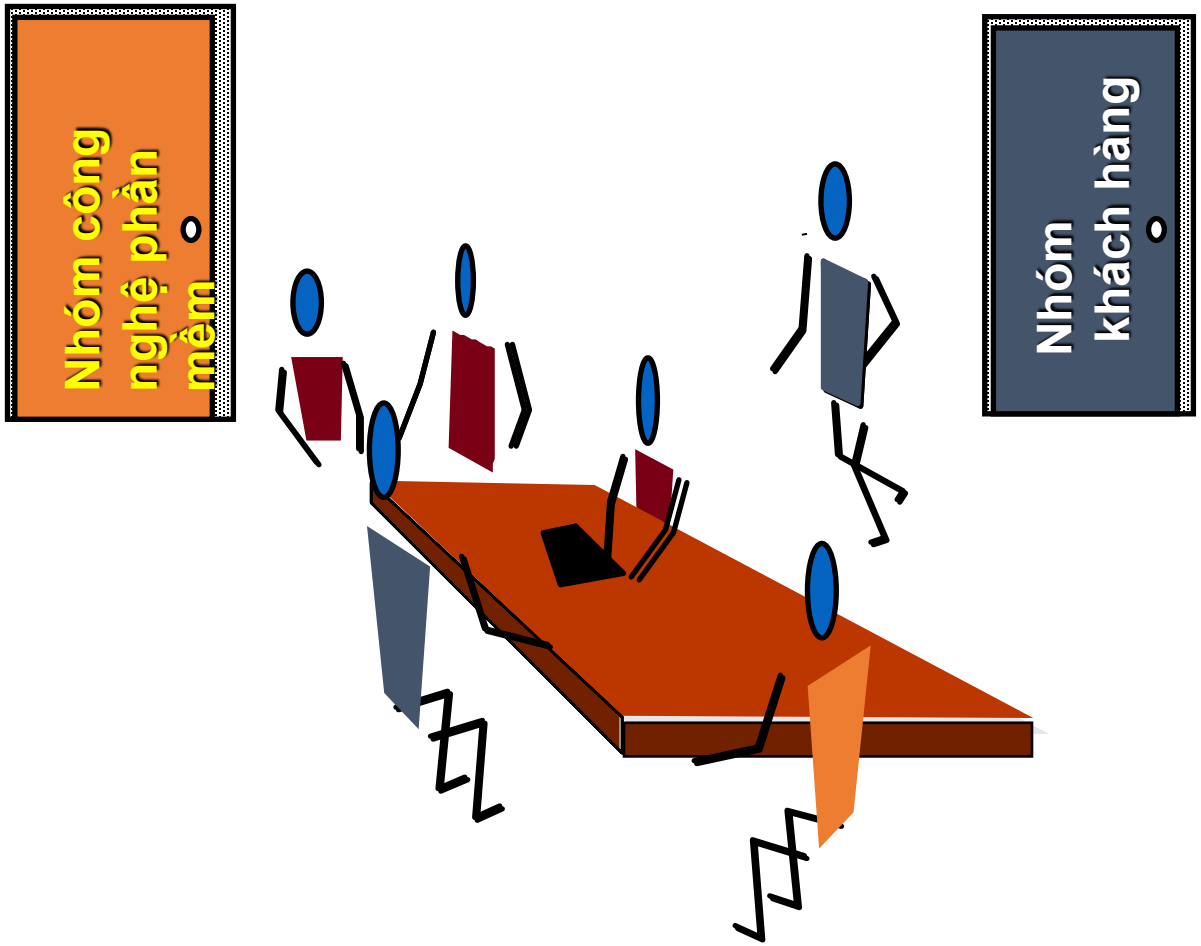
- Xác định các phương pháp sử dụng để phát hiện các yêu cầu phần mềm: phỏng vấn, làm việc nhóm, các buổi họp, gặp gỡ đối tác, v.v.
- Thu hút sự tham gia của nhiều chuyên gia, khách hàng để chúng ta có được các quan điểm xem xét phần mềm khác nhau từ phía khách hàng
- Xác định các yêu cầu còn nhập nhằng để làm mẫu thử
- Thiết kế các kịch bản sử dụng của phần mềm để giúp khách hàng định rõ các yêu cầu chính.

4a.Phát hiện yêu cầu phần mềm (tiếp)

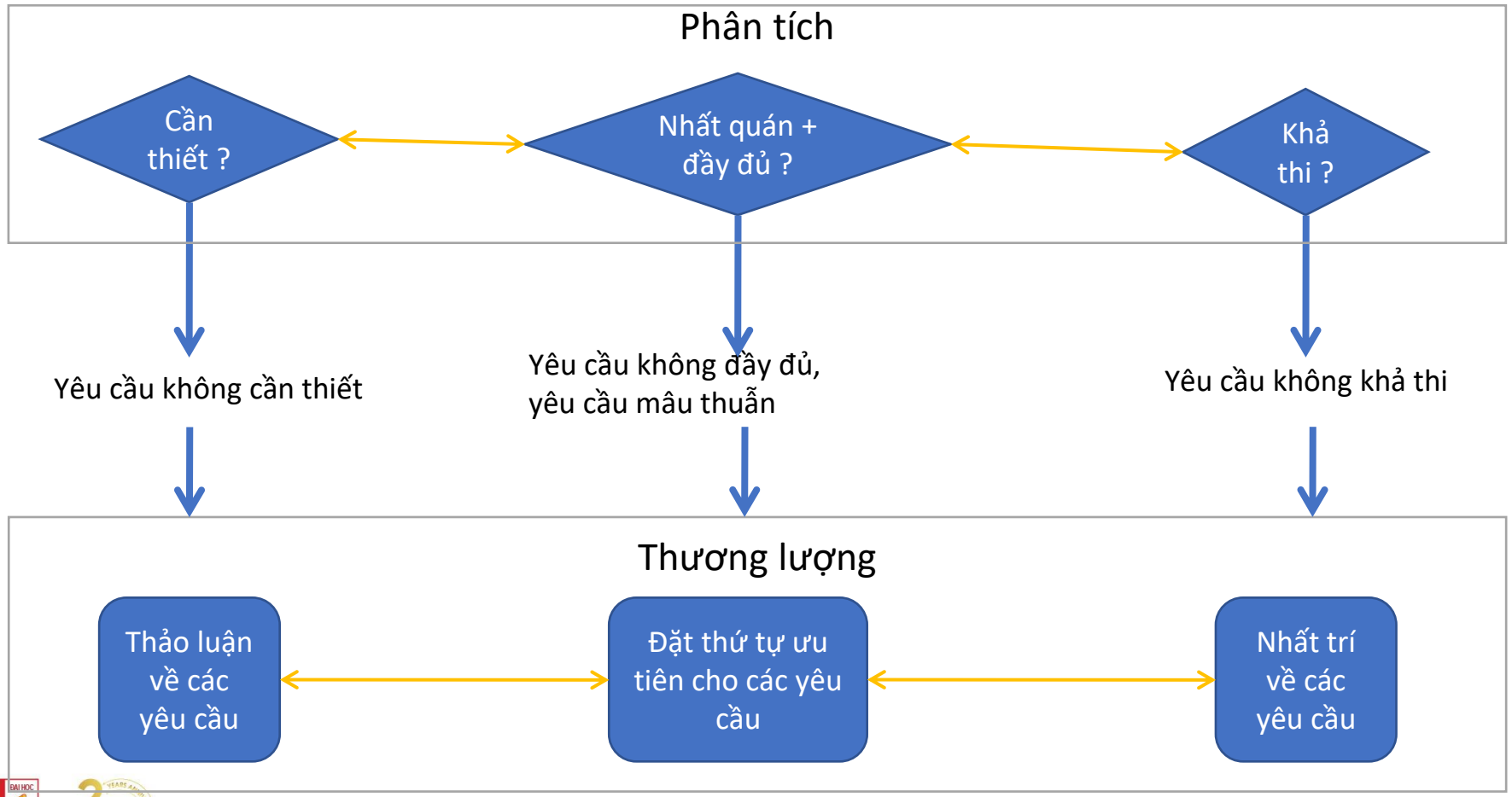
Đầu ra của bước phát hiện yêu cầu phần mềm

- Bảng kê (statement) các đòi hỏi và chức năng khả thi của phần mềm
- Bảng kê phạm vi ứng dụng của phần mềm
- Mô tả môi trường kỹ thuật của phần mềm
- Bảng kê tập hợp các kịch bản sử dụng của phần mềm
- Các nguyên mẫu xây dựng, phát triển hay sử dụng trong phần mềm (nếu có)
- Danh sách nhân sự tham gia vào quá trình phát hiện các yêu cầu phần mềm - kể cả các nhân sự từ phía công ty- khách hàng

4b.Phân tích các yêu cầu và thương lượng với khách hàng



4b.Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (tiếp)



b. Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (tiếp)

- Phân loại các yêu cầu phần mềm và sắp xếp chúng theo các nhóm liên quan
- Khảo sát tỉ mỉ từng yêu cầu phần mềm trong mối quan hệ của nó với các yêu cầu phần mềm khác
- Thẩm định từng yêu cầu phần mềm theo các tính chất: phù hợp, đầy đủ, rõ ràng, không trùng lặp
- Phân cấp các yêu cầu phần mềm theo dựa trên nhu cầu và đòi hỏi khách hàng / người sử dụng
- Thẩm định từng yêu cầu phần mềm để xác định:
  - Các yêu cầu PM có khả năng thực hiện được trong môi trường kỹ thuật hay không
  - Có khả năng kiểm định các yêu cầu phần mềm hay không

b.Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (tiếp)

- Thẩm định các rủi ro có thể xảy ra với từng yêu cầu phần mềm
- Đánh giá thô (tương đối) về giá thành và thời gian thực hiện của từng yêu cầu phần mềm trong giá thành sản phẩm phần mềm và thời gian thực hiện phần mềm
- Giải quyết tất cả các bất đồng về yêu cầu phần mềm với khách hàng/người sử dụng trên cơ sở thảo luận và thương lượng các yêu cầu đề ra

4c. Đặc tả yêu cầu phần mềm

- Đặc tả các yêu cầu phần mềm: xây dựng các tài liệu đặc tả, trong đó có thể sử dụng tới các công cụ như: mô hình hóa, mô hình toán học hình thức (a formal mathematical model), tập hợp các kịch bản sử dụng, các nguyên mẫu hoặc bất kỳ một tổ hợp các công cụ nói trên
- Phương pháp đặc tả:
  - Đặc tả phi hình thức (Informal specifications): viết bằng ngôn ngữ tự nhiên
  - Đặc tả hình thức (Formal specifications): viết bằng tập các ký pháp có các quy định về cú pháp (syntax) và ngữ nghĩa (sematic) rất chặt chẽ, thí dụ ký pháp đồ họa dùng các lưu đồ.
- Tiêu chí đánh giá chất lượng của hồ sơ đặc tả:
  - Tính rõ ràng, chính xác
  - Tính phù hợp
  - Tính đầy đủ, hoàn thiện

Ví dụ: Các yêu cầu về hồ sơ đặc tả

- Đặc tả hành vi bên ngoài của HT
- Đặc tả các ràng buộc về cài đặt
- Dễ thay đổi
- Dùng như công cụ tham khảo cho bảo trì
- Sự ghi chép cẩn thận về vòng đời của HT, nghĩa là dự đoán các thay đổi
- Các đáp ứng với các sự cố không mong đợi

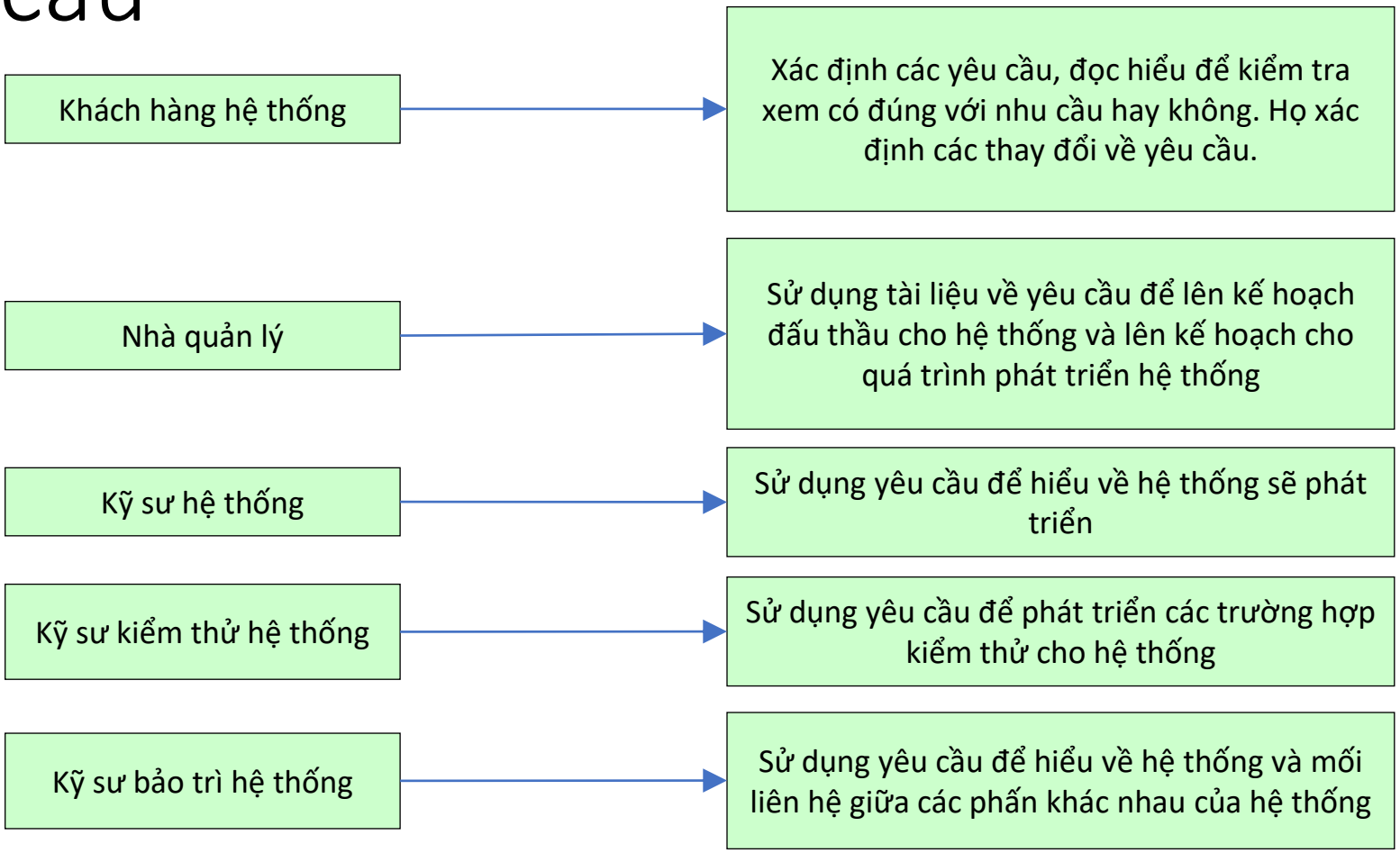
Các thành phần của hồ sơ đặc tả

- Đặc tả vận hành hay đặc tả chức năng (Operational specifications): mô tả các hoạt động của hệ thống phần mềm sẽ xây dựng:
  - Các dịch vụ mà hệ thống phải cung cấp
  - Hệ thống sẽ phản ứng với đầu vào cụ thể ra sao
  - Hành vi của hệ thống trong các tình huống đặc biệt.
- Đặc tả mô tả hay đặc tả phi chức năng (Descriptive specifications): đặc tả các đặc tính, đặc trưng của phần mềm:
  - Các ràng buộc về các dịch vụ hay các chức năng hệ thống cung cấp như thời gian, ràng buộc về các quá trình phát triển, các chuẩn,…
- Ngoài ra còn có yêu cầu về lĩnh vực, bắt nguồn từ lĩnh vực của ứng dụng hệ thống và các đặc trưng của lĩnh vực này.

Tài liệu yêu cầu

- Tài liệu về yêu cầu là các phát biểu chính thức về cái được yêu cầu bởi các nhà phát triển hệ thống
- Nó bao gồm cả 2 phần:
  - định nghĩa và đặc tả yêu cầu
- Nó không phải là tài liệu thiết kế. Tốt hơn có thể nó chỉ là 1 tập các cái mà hệ thống phải làm hơn là hệ thống phải làm thế nào (phân tích chứ không phải là thiết kế)

Nội dung cần có của tài liệu yêu cầu

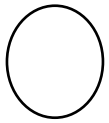


4d. Một số mô hình hóa hệ thống

- Biểu đồ phân cấp chức năng - WBS (work break down structure)
- Biểu đồ luồng dữ liệu – DFD (data flow diagram)
- Máy trạng thái – FSM (Finite state machine)
- Sơ đồ thực thể liên kết – ERD (entity relation diagram)

Ví dụ: Đặc tả chức năng với DFD

- Hệ thống (System): tập hợp các dữ liệu (data) được xử lý bằng các chức năng tương ứng (functions)
- Các ký pháp sử dụng:



Thể hiện các chức năng (functions)

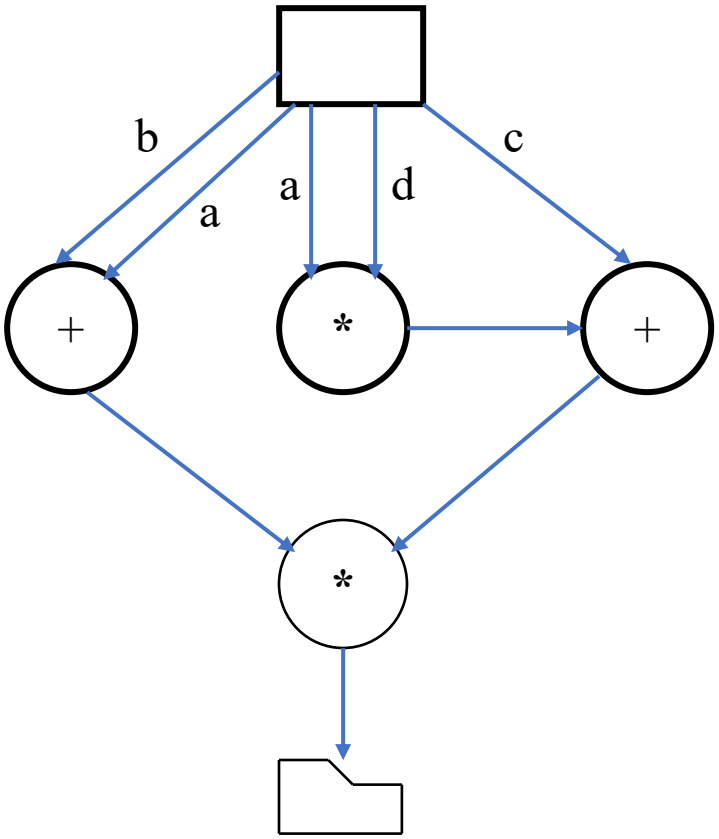
Thể hiện luồng dữ liệu

Kho dữ liệu

Vào ra dữ liệu và tương tác giữa hệ thống và người sử dụng

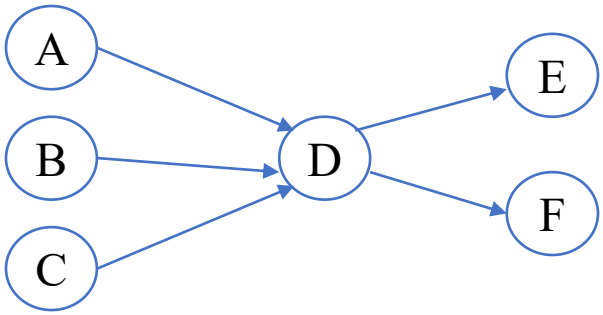
Ví dụ: mô tả biểu thức toán học bằng DFD

(a+b)*(c+a*d)-e*(a+b)



Các hạn chế của DFD

- Trong DFD không xác định rõ các hướng thực hiện (control aspects)

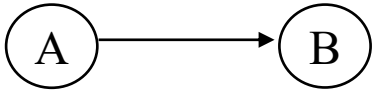


- Biểu đồ DFD này không chỉ rõ đầu vào là gì để thực hiện chức năng D và đầu ra là gì sau khi thực hiện chức năng D.

- Chức năng D có thể cần cả A, B và C
- Chức năng D có thể chỉ cần một trong A, B và C để thực hiện
- Chức năng D có thể kết xuất kết quả cho một trong E và F
- Chức năng D có thể kết xuất kết quả chung cho cả E và F
- Chức năng D có thể kết xuất kết quả riêng cho cả E và F

Các hạn chế của DFD

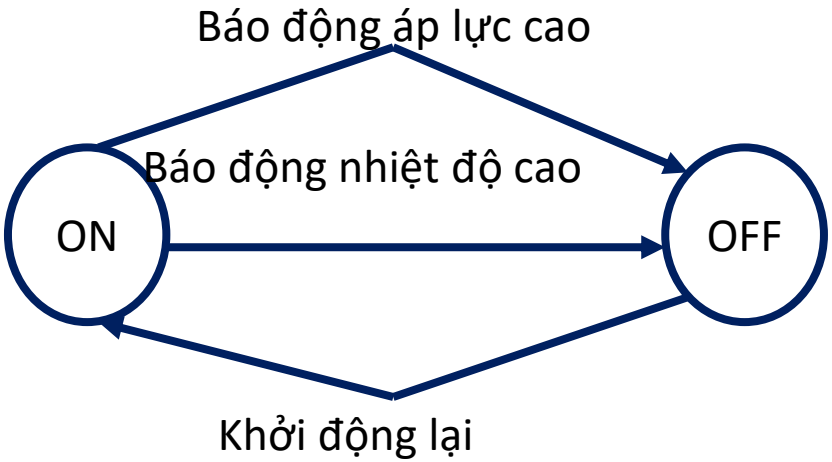
- DFD không xác định sự đồng bộ giữa các chức năng / mô-đun
  - A xử lý dữ liệu và B được hưởng (nhận) các kết quả được xử lý từ A
  - A và B là các chức năng không đồng bộ (asynchronous activities) vì thế cần có buffer để ngăn chặn tình trang mất dữ liệu



Ví dụ: Đặc tả trạng thái với FSM - Finite State Machines

- FSM chứa
  - Tập hữu hạn các trạng thái Q
  - Tập hữu hạn các đầu vào I
  - Các chức năng chuyển tiếp

$$\delta : Q \times I \rightarrow Q$$



Ví dụ: Đặc tả dữ liệu với Mô hình thực thể liên kết -ERD

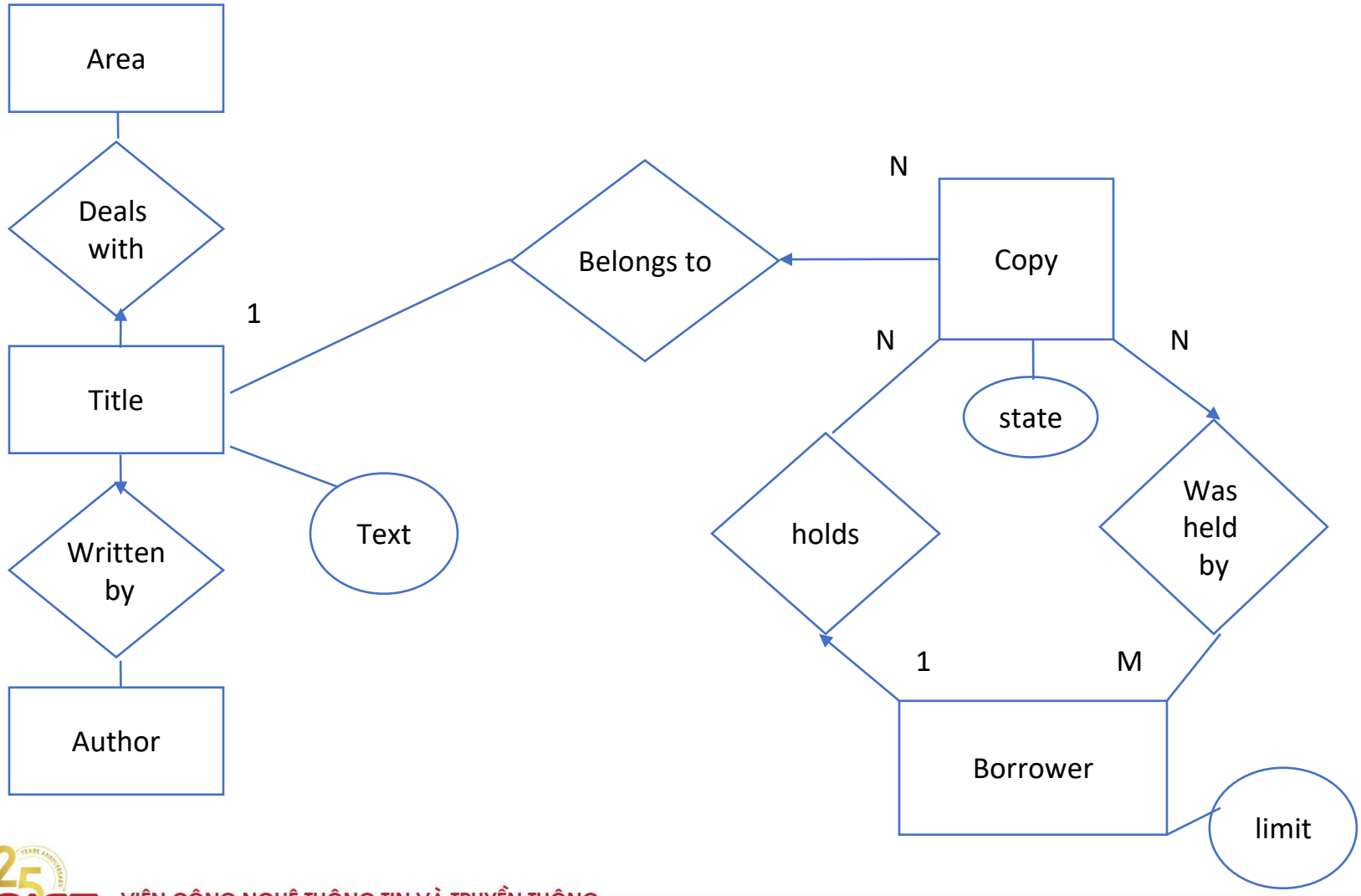
- Mô hình khái niệm cho phép đặc tả các yêu cầu logic của hệ thống, thường được sử dụng trong các hệ thống dữ liệu lớn
- ER Model
  - Thực thể
  - Quan hệ
  - Thuộc tính
- Biểu đồ thực thể

Thực thể

- Thực thể : tập hợp các thông tin liên quan cần được xử lý trong phần mềm
- Thực thể có thể có mối quan hệ:
  - người sở hữu ôtô

- Thực thể có các thuộc tính

Ví dụ: ERD mô tả thư viện



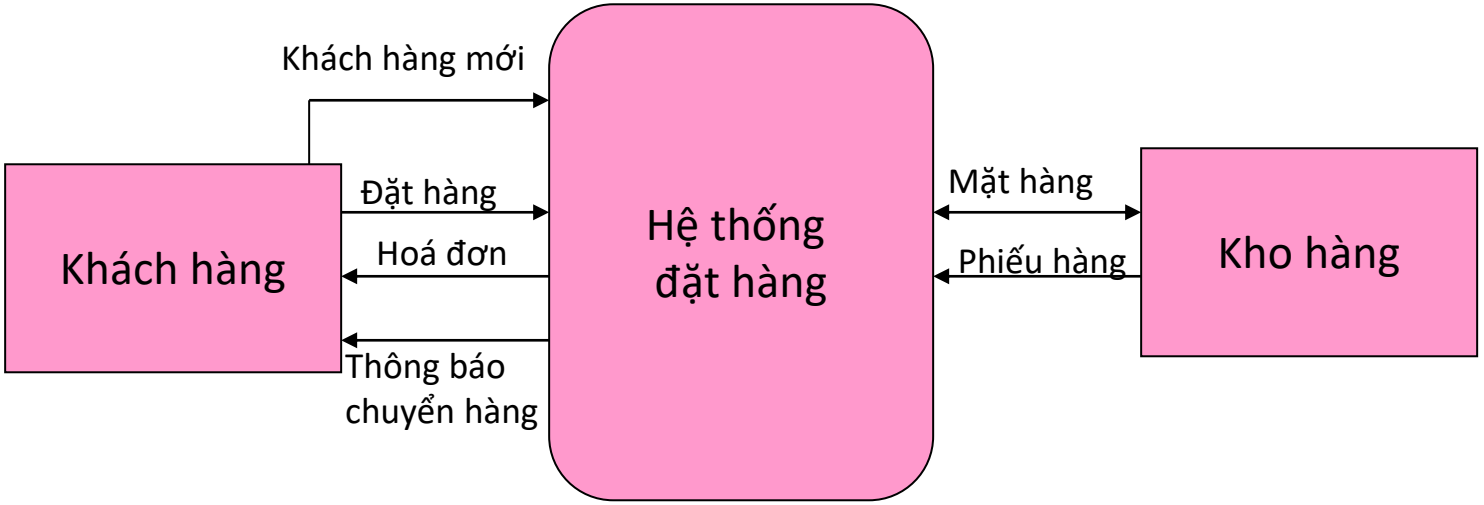
So sánh

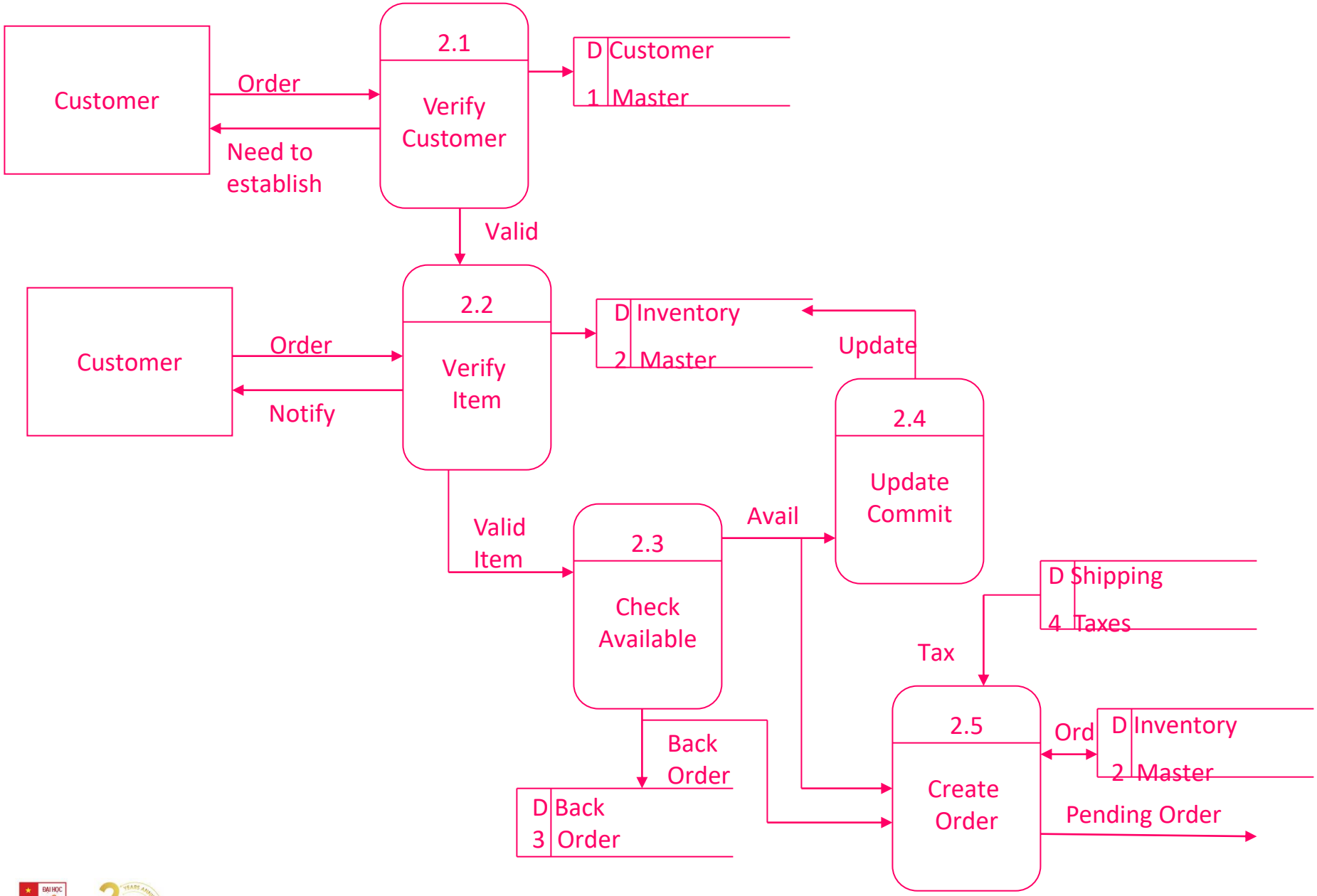
DFD	FSM	ERD
Đơn giản, dễ hiểu.	Có thể phức tạp với số lượng trạng thái lớn	Đơn giản, dễ hiểu
Mô tả luồng dữ liệu	Mô tả trạng thái của thực thể	Mô tả trừu tượng cơ sở dữ liệu
Không xác định rõ hướng thực hiện	Xác định rõ hướng thực hiện	Không xác định rõ hướng thực hiện
Không thể hiện tính tuần tự hay song song của tiến trình	Thể hiện tốt tính song song và tuần tự	Không thể hiện tính tuần tự hay song song

Thế nào là một đặc tả tốt?

- Dễ hiểu với người dùng
- Có ít điều nhập nhằng
- Có ít quy ước khi mô tả, có thể tạo đơn giản
- Với phong cách từ trên xuống (topdown)
- Dễ triển khai cho những pha sau của vòng đời:
  - thiết kế hệ thống, thiết kế chương trình và giao diện dễ làm, đảm bảo tính nhất quán, . . .

Thế nào là một đặc tả tốt?




Customer
Order
Need to establish
2.1
Verify Customer
D 1 Customer Master
Valid
Customer
Order
Notify
2.2
Verify Item
D 2 Inventory Master
Update
2.4
Update Commit
Valid Item
2.3
Check Available
Avail
D 4 Shipping Taxes
Tax
2.5
Create Order
Ord
D 2 Inventory Master
Back Order
D 3 Back Order
Pending Order

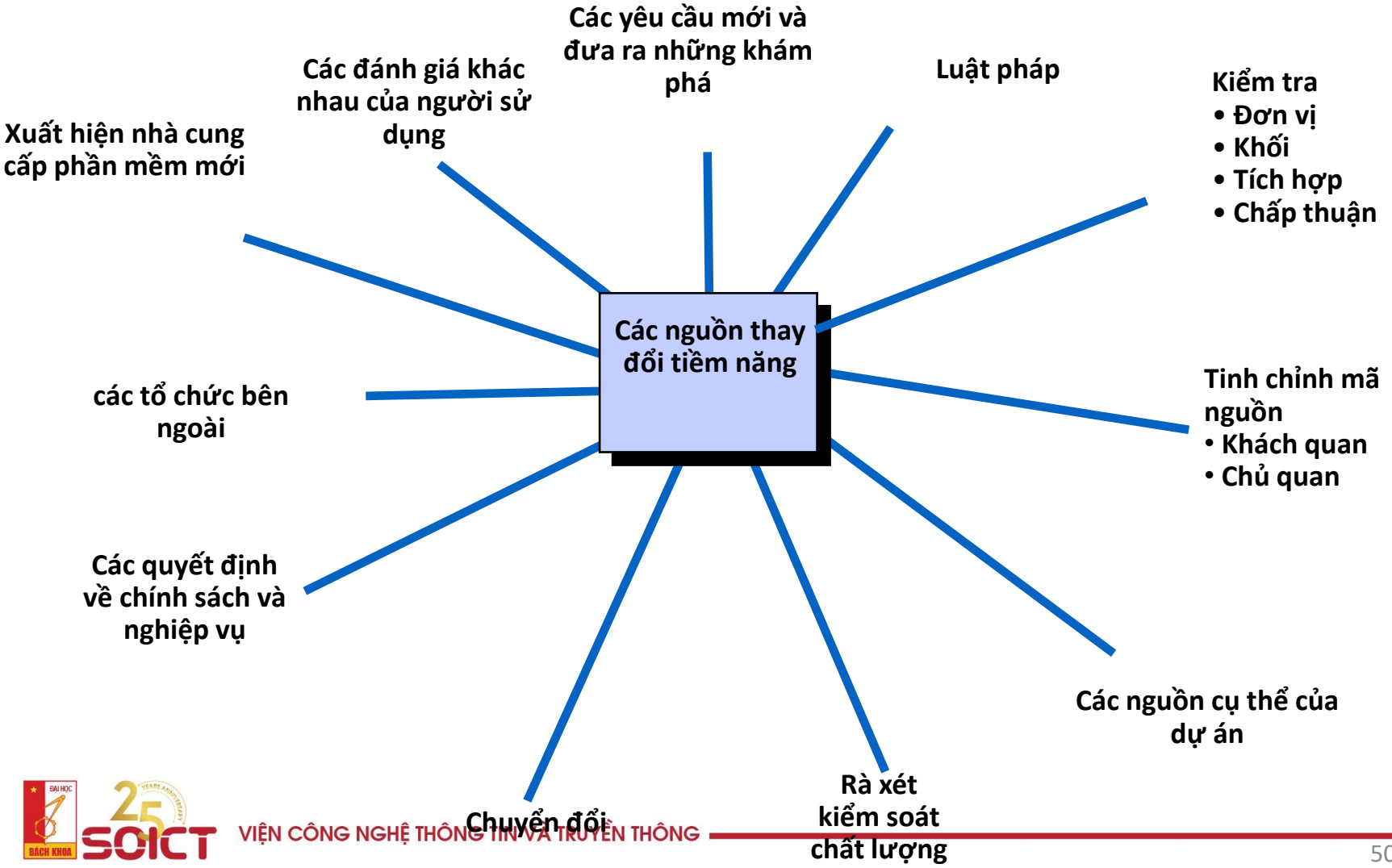
4e. Quản trị các yêu cầu phần mềm

- Các công việc liên quan
  - ✓Quản lý thay đổi và vấn đề phát sinh
  - ✓Kiểm soát nguồn thay đổi tiềm năng
- Lợi ích:
  - ✓Chi phí phát triển thấp hơn trong suốt vòng đời phát triển phần mềm
  - ✓Ít sai sót hơn
  - ✓Giảm thiểu rủi ro đối với các sản phẩm quan trọng về an toàn
  - ✓Giao hàng nhanh hơn
  - ✓Khả năng tái sử dụng
  - ✓Truy xuất nguồn gốc
  - ✓Các yêu cầu gắn liền với các trường hợp thử nghiệm

Quản lý thay đổi và vấn đề phát sinh

- Thay đổi là gì ?
  - Bất cứ hoạt động nào thay đổi phạm vi, kết quả bàn giao, kiến trúc cơ bản, chi phí, lịch trình của một dự án
- Tại sao cần phải quản lý thay đổi và vấn đề phát sinh ?
  - Thay đổi và vấn đề phát sinh là 2 lý do thường làm dự án thất bại
- Làm thế nào để kiểm soát thay đổi và giải quyết các vấn đề phát sinh ?
  - Giảm rủi ro dự án nhờ quy trình hiệu quả quản lý thay đổi và vấn đề
  - Các thành viên nhóm hiểu được quy trình quản lý thay đổi và vấn đề
  - Ghi chép đầy đủ về các yêu cầu thay đổi/ vấn đề

Kiểm soát nguồn thay đổi tiềm năng



Tổng kết

1. Sau bài học, sinh viên đã nắm được các kĩ năng:

- Các khái niệm liên quan tới kĩ nghệ YCPM
- Các hoạt động chính trong kĩ nghệ YCPM
- Một số mô hình thường dùng trong quá trình đặc tả YCPM

2. Trong bài tiếp theo sẽ giới thiệu tổng quan về các mô hình trong kĩ nghệ YCPM

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict